CS6640-Project 2

Atul Rungta
U0734304
atul.rungta@utah.edu

1. Image Smoothing

a. K*K smoothing kernel with equal weights

A cross-convolution scheme was implemented using a normalized filter (3x3 and 5x5). The matrix chosen was all ones (normalized according to dimensions). The results of smoothing:

Original Image

5x5 Smoothing Filter

3x3 Smoothing Filter

b. Separable Filter

The box filter used was separated into 2 unit vectors (of the appropriate dimension) and normalized accordingly. The image was first convolved with the horizontal filter followed by the vertical one. The results:

3x3 separable filter

5x5 separable filter

Although the results should match the ones for the box filter, there seems to be a slight difference in the intensity of the image. My guess would be that it is happening because of I am using convolution in case of separable filters while cross-correlation in case of box filters.

c. Separable Gaussian filter

The 2D Gaussian:

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-[(x-\mu_x)^2+(y-\mu_y)^2]/(2\sigma^2)},$$

where σ^2 = variation, μ_x, μ_y are the means.

Can be separated into 2 1D Gaussians as:

f(x) = (1/((2 * pi) ^ .5 * σ)) * e ^ (-(x-μ)^2/(2 σ²))
f(y) = (1/((2 * pi) ^ .5 * σ)) * e ^ (-(y-μ)^2/(2 σ²))

σ was chosen to be 2, μ was 0 and ranges ±3 σ were chosen. The results:

Separable Gaussian Filter (± 3σ)

The resulting image yields smoother image as compared to a 5x5 box filter.

2. Edge Detection

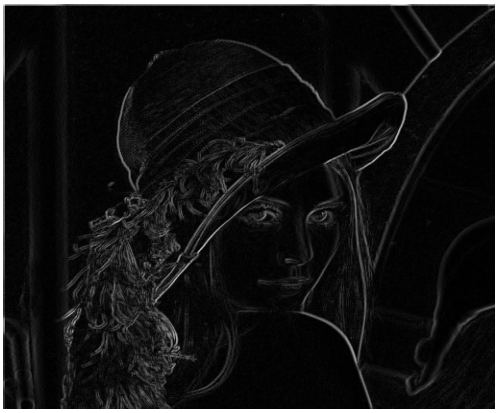
a. Local Edge Filtering

For the horizontal filtering, dx was chosen to be [-1 0 1] while dy was chosen to be $[-1\ 0\ 1]^T$. The masks were not normalized as it killed the intensity. The results were obtained by convolving the mask with the image. The result

Dx

Dy

Edgemap



Orientation Map

For the orientation map, the arctan values were converted to degrees and adjusted for values between 0 to 255.

b. Edge Filtering at specific scales

As mentioned in the assignment, the image was filtered using the Gaussian (σ =2) and then the edge map was calculated. The result is a much smoother edgemap as expected:

3. Template Matching

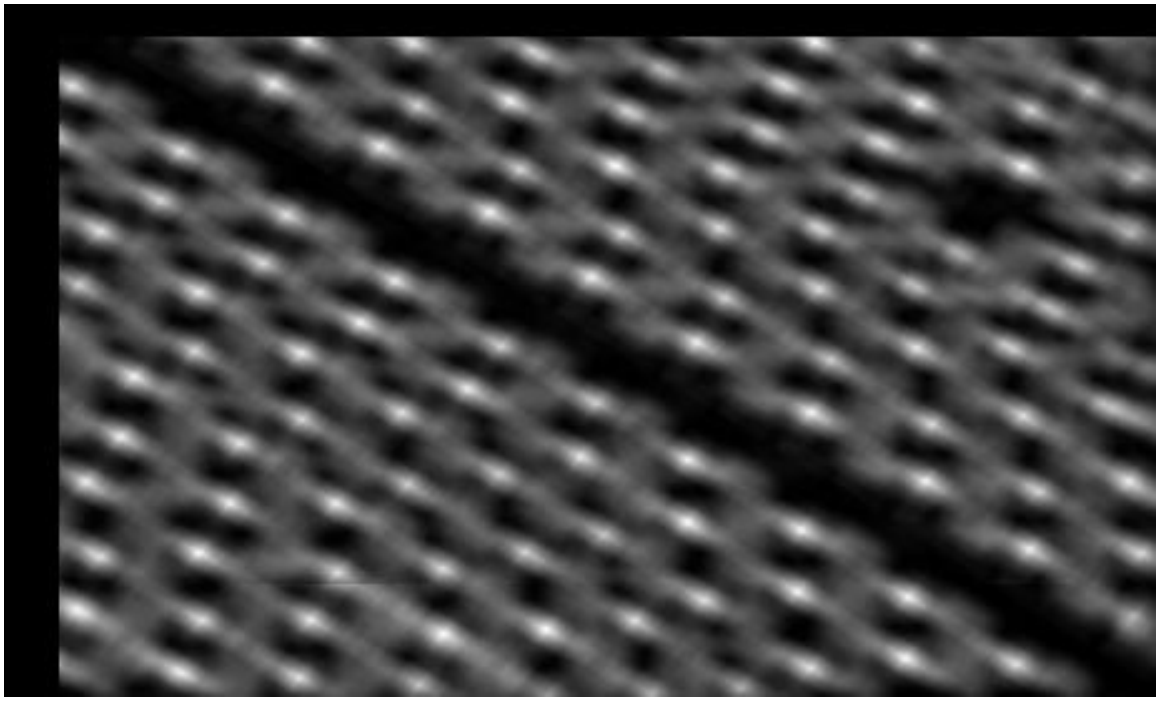
The template matching didn't quiet work as expected. As explained, a small segment (template) was chosen and made the filter. The image was cross-correlated with the filter and the resulting image was thresholded. The following equation was used for the cross-correlation (taken from http://paulbourke.net/miscellaneous/correlate/)

$$r[i][j] = \sum_{jj=-Nj/2}^{jj<Nj/2} \sum_{ii=-Ni/2}^{ii<Ni/2} (mask[ii+Ni/2][jj+Nj/2] - \overline{mask})(image[i+ii][j+jj] - \overline{image})$$

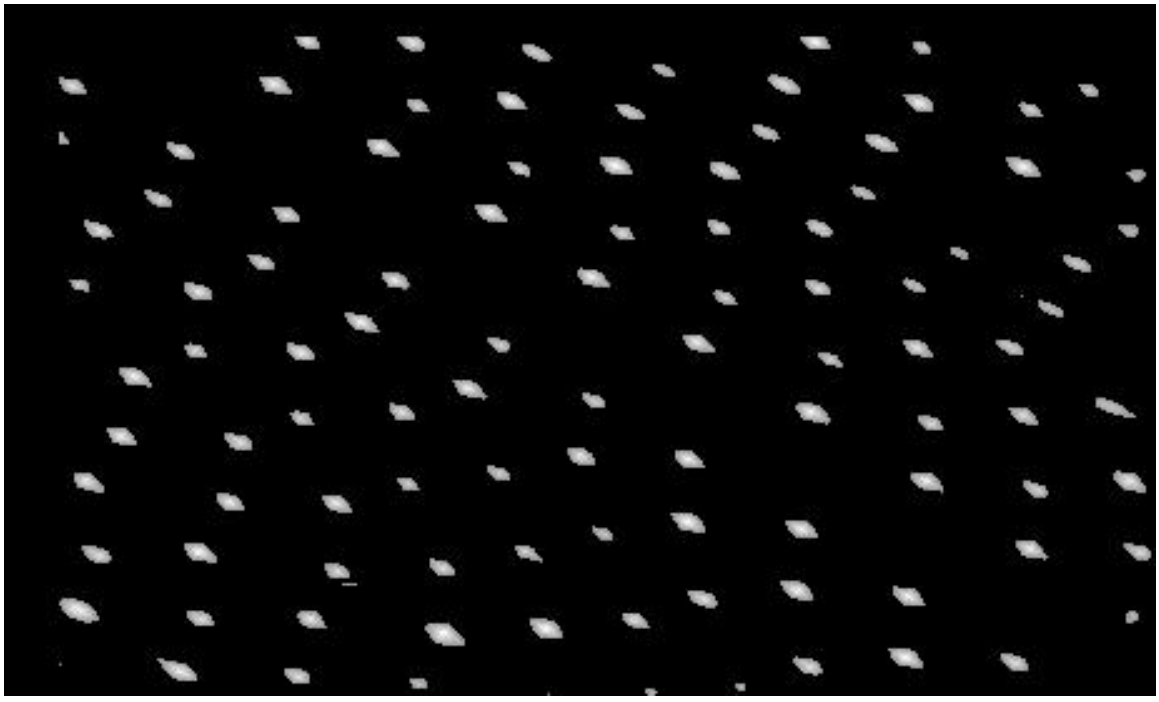
where mask' is the mean of the template and image' is the mean of the image raster under the mask.

The threshold limit was set to 200 by looking at the histograms of the correlation images. Results:

Original Image



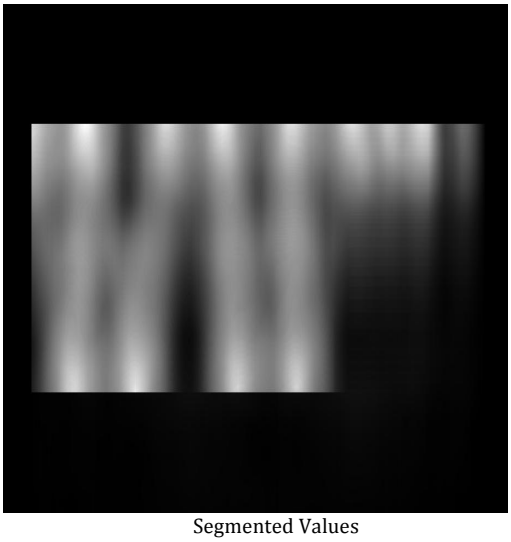
Segmented



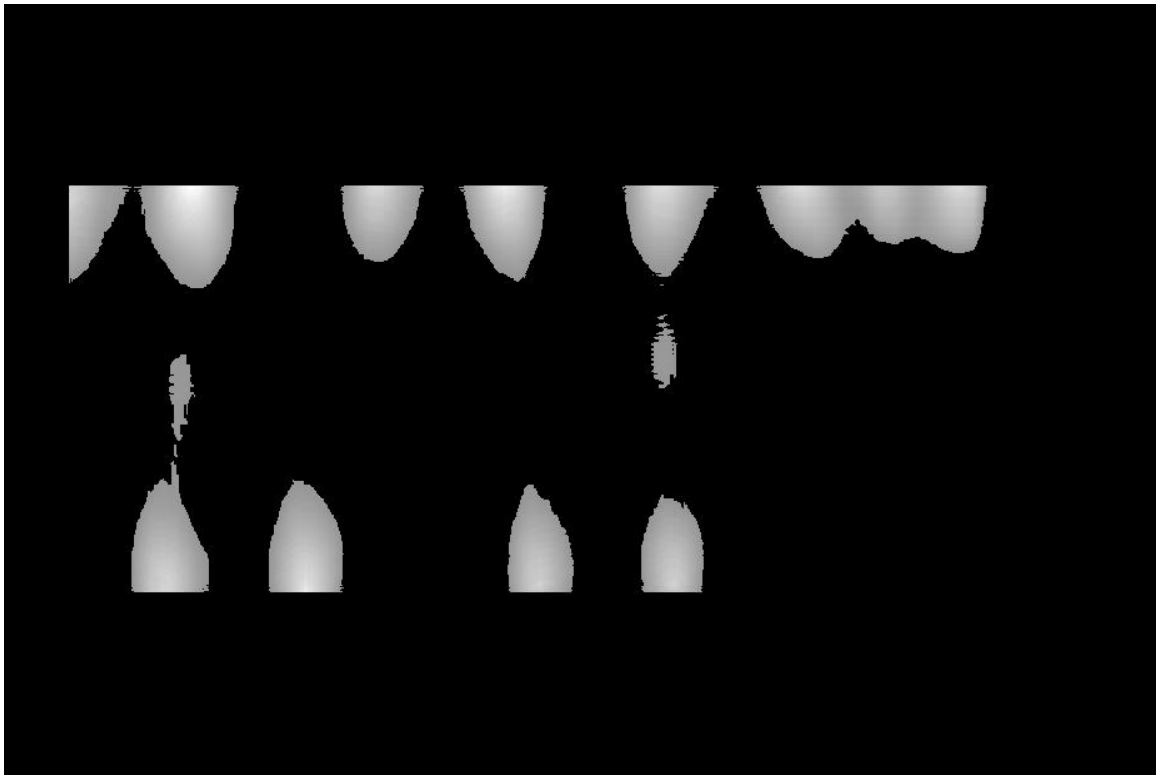
Thresholded Values

Original Image

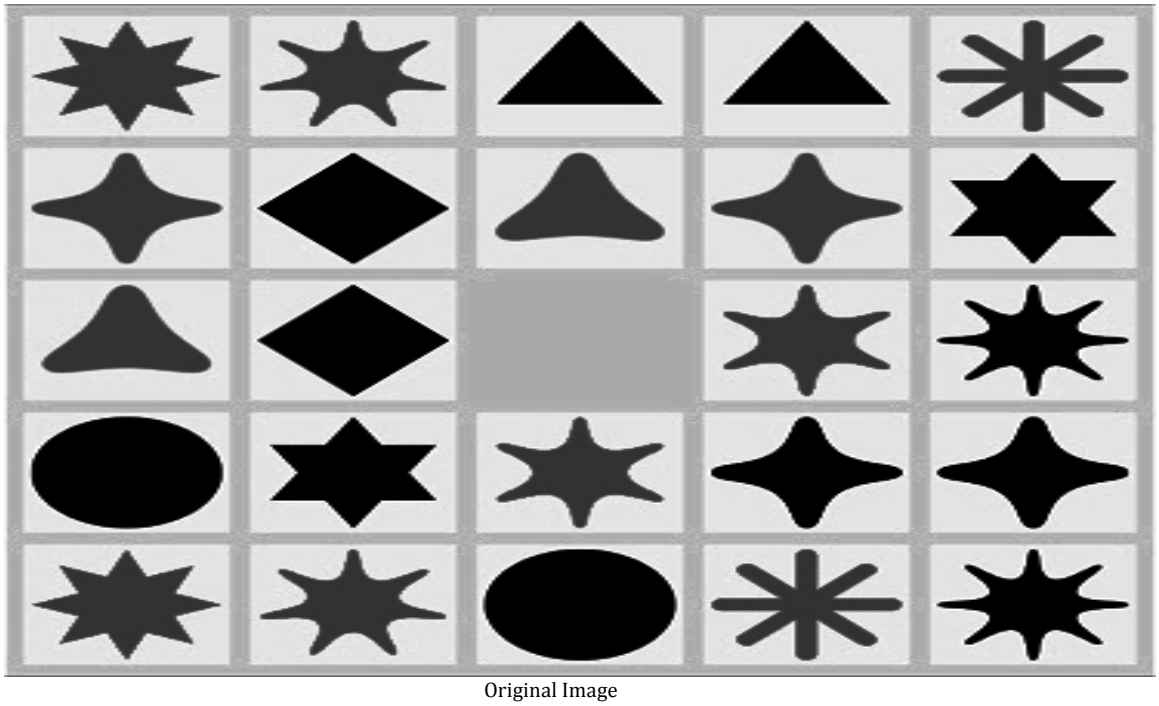
Key Segment



Segmented Values

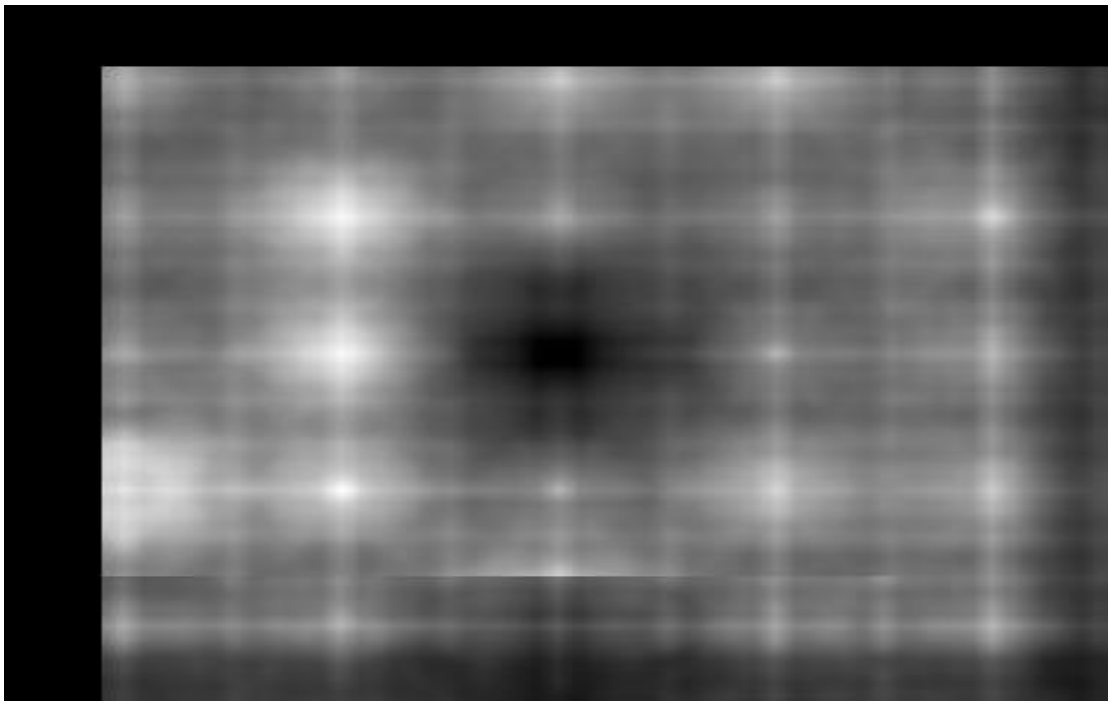


Thresholded Values

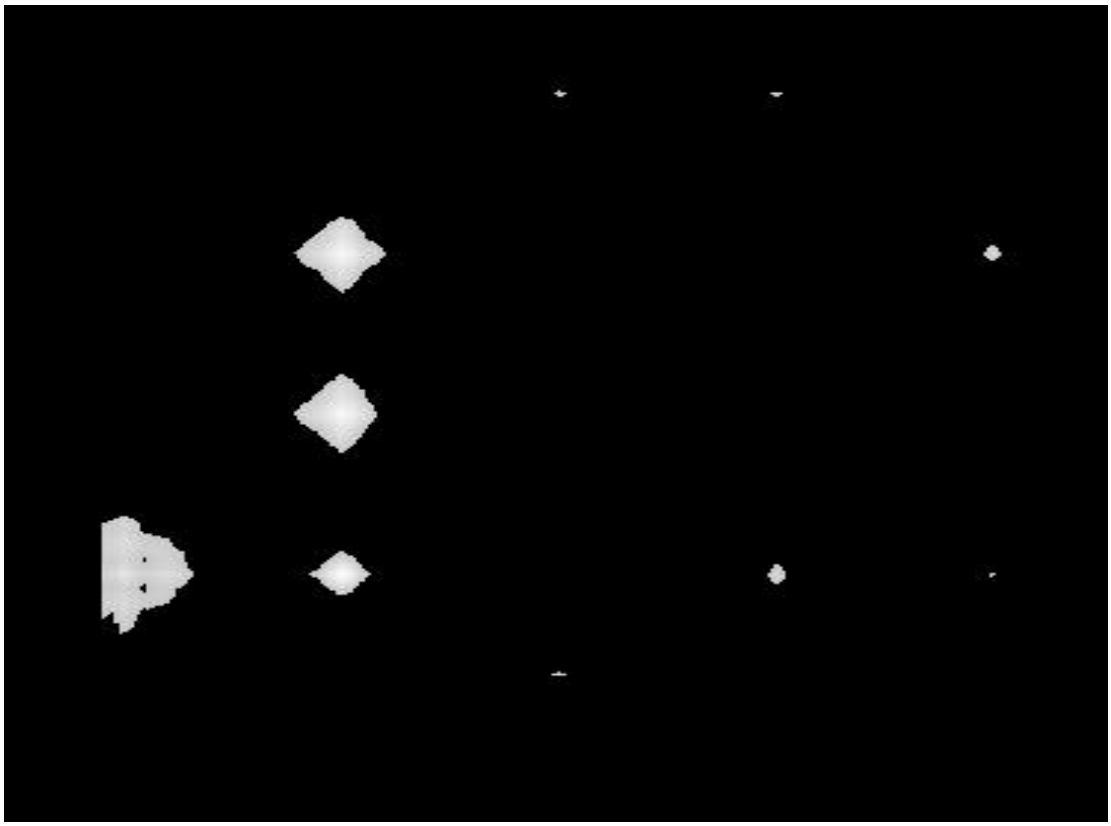


Original Image

Segment



Segmented Values



Thresholded Values (These are incorrect)

Template matching seems to work well in a few cases, but as seen above it gives the wrong answer (this could also be a glitch in my code).